Veronte MC280

Embention

Nov 05, 2021

CONTENTS

This document describes the main functionalities of the MC280 Speed Controller.

Veronte MC280 is capable to drive a wide range 3-phase PMSM motors as well as brushless motors.The MC280 uses a FOC algorithm to control the motor.

The MC280 Speed Controller has an input voltage range from 12-75V with a maximum continuous current of 280A.

The system has a temperature range from -40 to 65ºC. For higher temperature, a power limitation will be applied.

CHAPTER

ONE

HARDWARE DESCRIPTION

1.1 MC280 Specifications

Table 1: Electrical Characteristics

Type	Specification
Voltage	12-75VDC
Cont. Current	5 - 280A
Peak Current (<5s)	300A
Sensorless Motor	Yes
Cooling	Air (Fan optional)
Redundant Control	Yes
Regenerative Brake	Yes
Maximum speed (2 pole)	100000
Weigth	740g (without cables)

1.2 Main Scheme

The system consist in: - 2 Battery cables - 1 Interface connector - 3 Motor cables

1.3 Installation Considerations

When installing the MC280 speed controller in the vehicle, the following limitation shall be considered:

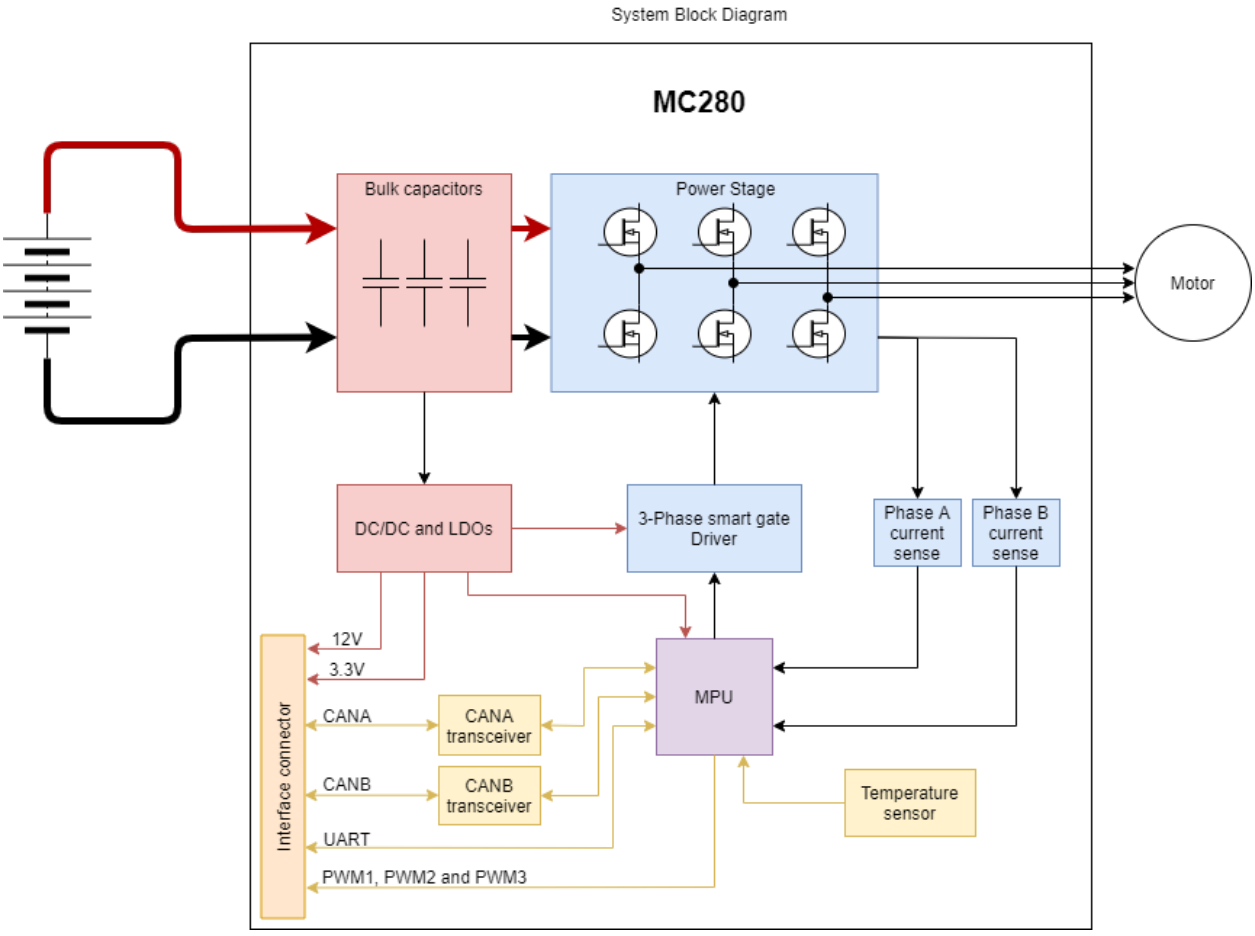
- The maximum distance between **battery/power supply and MC280 shall be 1m**
- The maximum distance between **the Controller System and the motor is 1m**
- The wire connection type between the power items must be crimped not soldered.
- The system must be placed in a ventilated place with proper air flow. If this is not possible, it is necessary to install an external fan.

Note: When working voltage is higher than 60V, use of insulating gloves is mandatory and the system **must have** a chassis fault detection system.

Warning:  **Careful!** The system does not discharge the voltage on the input terminals when the battery is disconnected. Capacitors remain charged.

1.4 Function description

The MS280 system block diagram is shown below.



The battery input voltage is first introduced to a bulk capacitor stage for and EMI improvement and to reduce input ripple.

After the Bulk Capacitor stage, the input voltage is applied to the power stage and the DCDC / LDO converters to power the MPU and the rest of the peripherals.

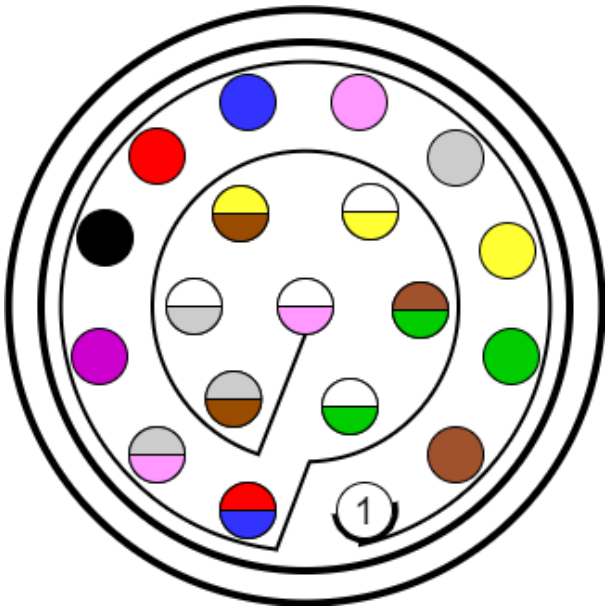
The motor is controlled via CAN.The UART TTL port is mainly used for telemetry reading and for firmware update.The PWM / ECAP can be configured and used for connecting, for example a quadrature encoder, fan speed control, PPM input. . . All these buses is accessible in the interface connector.

An internal temperature sensor provides the PCB temperature to the MPU.

The MPU controls the smart driver, who is in charge of the MOSFETs activation. With the current sensor located in phase A and phase B the MPU can determine the position of the motor at any time.

1.5 Interface Connector Pinout

The Interface connector pinout is shown in the following table:



Main Connector

Table 2: Interface Connector

Pin	Signal	Type	Comment
1	CANB -	CAN communication	No internal bus termination resistor
2	CANA +	CAN communication	No internal bus termination resistor
3	CANB +	CAN communication	No internal bus termination resistor
4	CANA -	CAN communication	No bus internal termination resistor
5	PWM1/ECAP	Digital GPIO	Configurable I/O
6	PWM2/ECAP	Digital GPIO	Configurable I/O
7	GND	Digital Ground	
8	PWM3/ECAP	Digital GPIO	Configurable I/O
9	UART-RX	UART Input	
10	UART-TX	UART Output	
11	12V	12V power output	No fuse protected
12	3.3V	3.3V power output	No fuse protected
13	GND	Digital Ground	
14	GND	Digital Ground	
15	N.C.	Not connected	
16	N.C.	Not connected	
17	N.C.	Not connected	
18	N.C.	Not connected	
19	N.C.	Not connected	

CAN

Differential communication protocol. This bus is used to command the MC280 motor driver as well as receive telemetry.

Table 3: Electrical Characteristics

Type	Specification
ESD Protection	±16 kV (HBM)
Requirements	ISO11898
Isolation	No
Speed	Max. 1 Mbit/s
Input Voltage(D)	-7V to 12V
Output Voltage (D)	2.45V to 3.3V

PWM/ECAP

Those signals can be configured as GPIO, PWM output, PPM input and ECAP input.

Table 4: Electrical Characteristics

Type	Specification
Max Input Voltage	3.3V
Max Output Voltage	3.3V
Max Output Drain	1.5mA

UART TTL

This port is used to output the thelemetry, MC280 configuration and firmwareupdate.

Table 5: Electrical Characteristics

Type	Specification
Input Voltage	3.3V
Output Voltage	3.3V

12V

12V output. Tipicaly used to provide power to cooling fan, external sensors...

Table 6: Electrical Characteristics

Type	Specification
Output Voltage	12V
Max Current Output	1A

Note: Do not use to supply loads higher then 1A.

3.3V

3.3V output.

Table 7: Electrical Characteristics

Type	Specification
Output Voltage	3.3
Max Current Output	500mA

Note: Do not use to supply loads higher then 500mA

1.6 Wiring definition

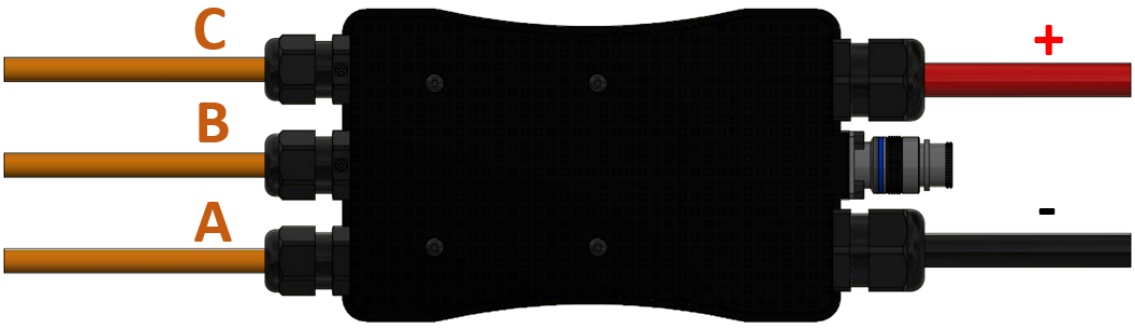
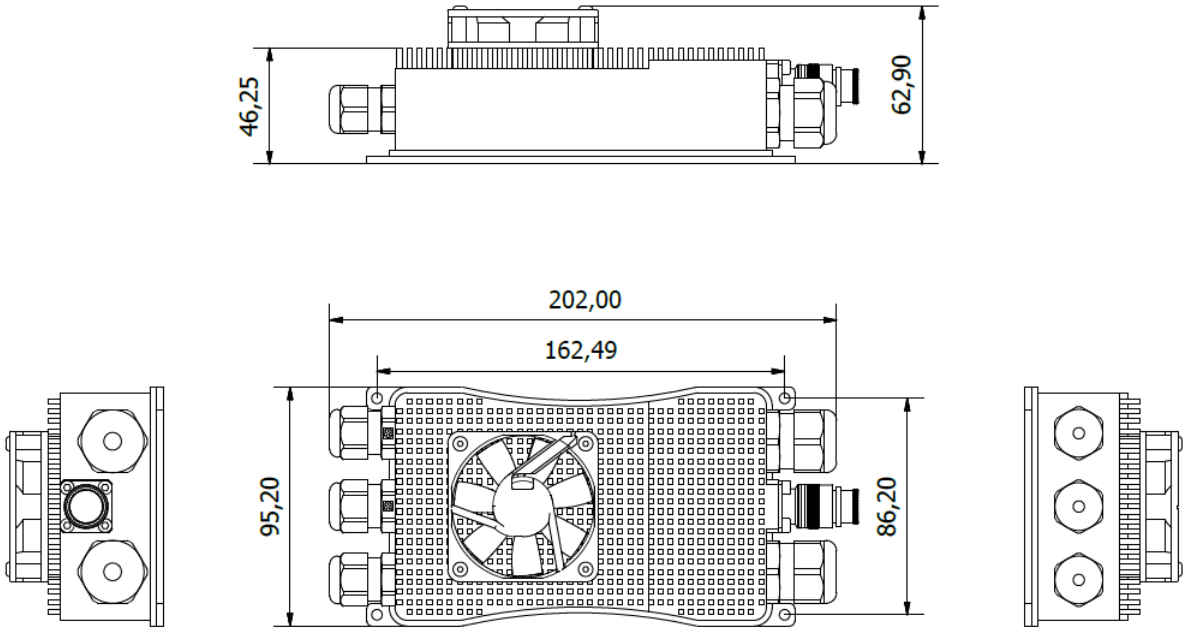


Table 8: Wire dimensions

Color	cross-sectional area	AWG
Red	35mm^2	2AWG
Black	35mm^2	2AWG
Orange	16mm^2	6AWG

1.7 Mounting Instructions

The MC280 motor controller has the following dimensions and fixation points. All fixation holes are D4.2mm ready to use with an M4 screw.



Mounting dimensions

CHAPTER

TWO

SOFTWARE SETUP

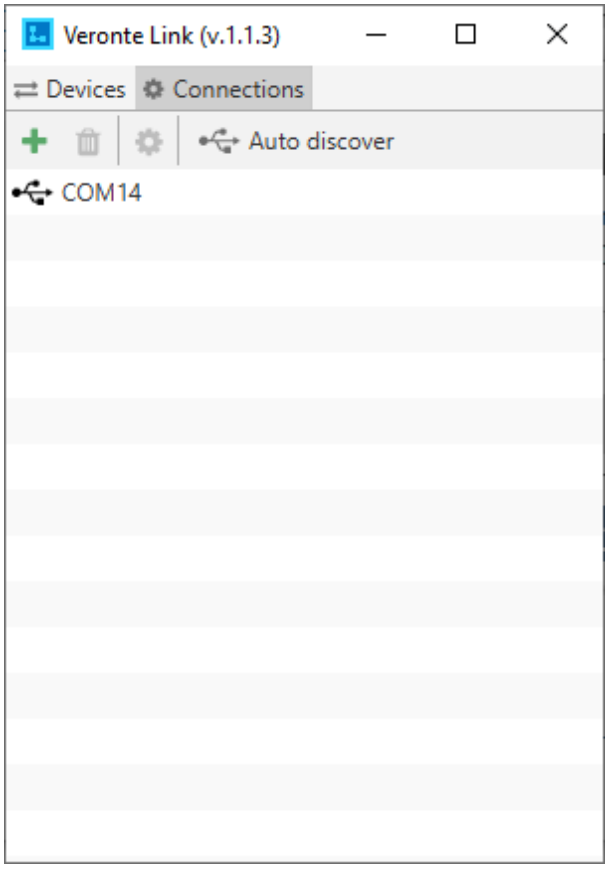
To properly connect to a MC280 unit you must previously install **Veronte Link and MC280 PDI Builder**.

1. **Veronte Link**: This tool is the HUB that manages all Veronte and Embention devices that use a COM port. Here the user can check, configure and list the devices that are currently connected.
2. **MC280 PDI Builder**: This tool is used to set all the configurable parameters. Here the user can set, tune and define the motor, control and sensors that are going to be used alongside the ESC.

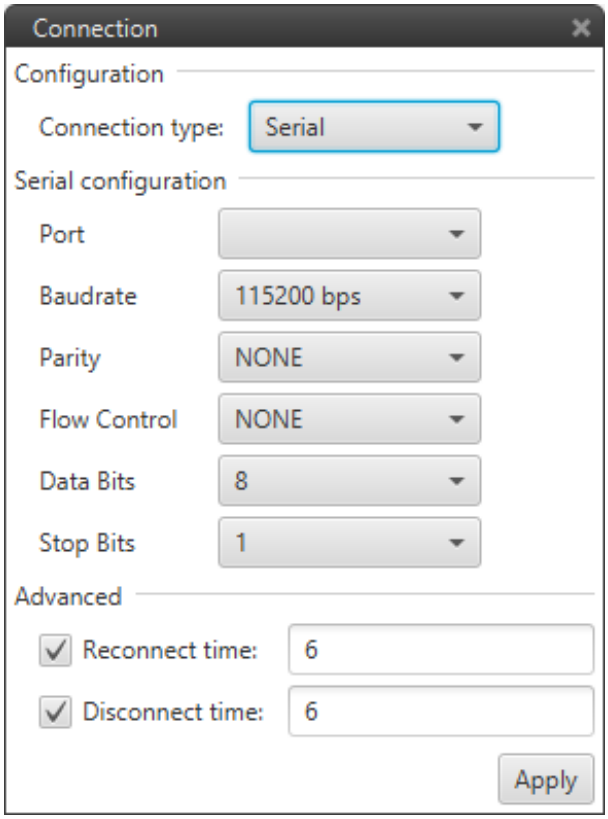
2.1 Veronte Link Setup

Once Veronte Link is installed, the first step that must be done is to set the com port that your MC280 unit is currently using. By default, every MC280 is capable to comunicate though USB, RS232 and RS485 so any of theese can be used (properly adapted to USB/serial).

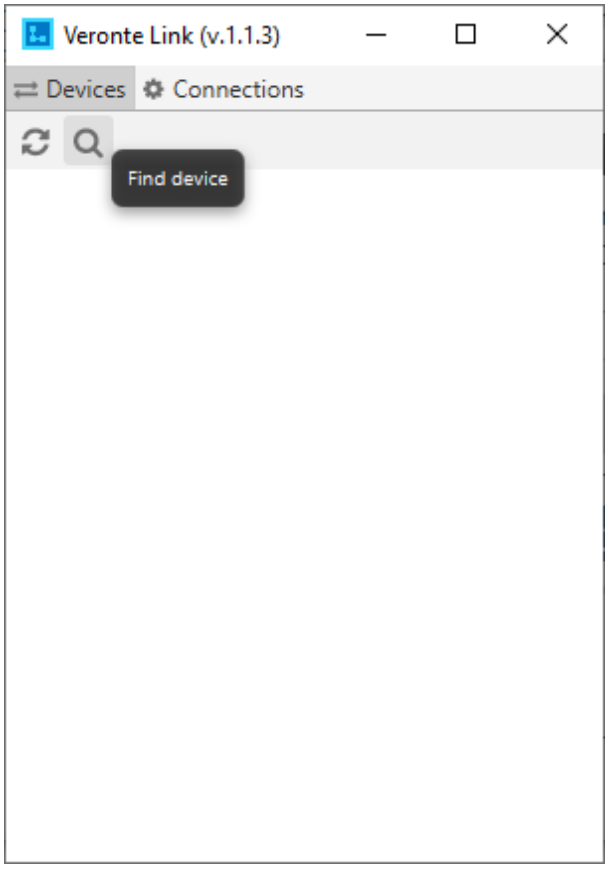
First, click on "**+**":



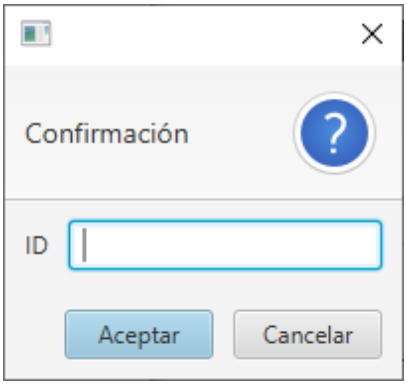
Select your COM settings, typically, those shown below:



After that, go to devices tab and click on **find devices**:



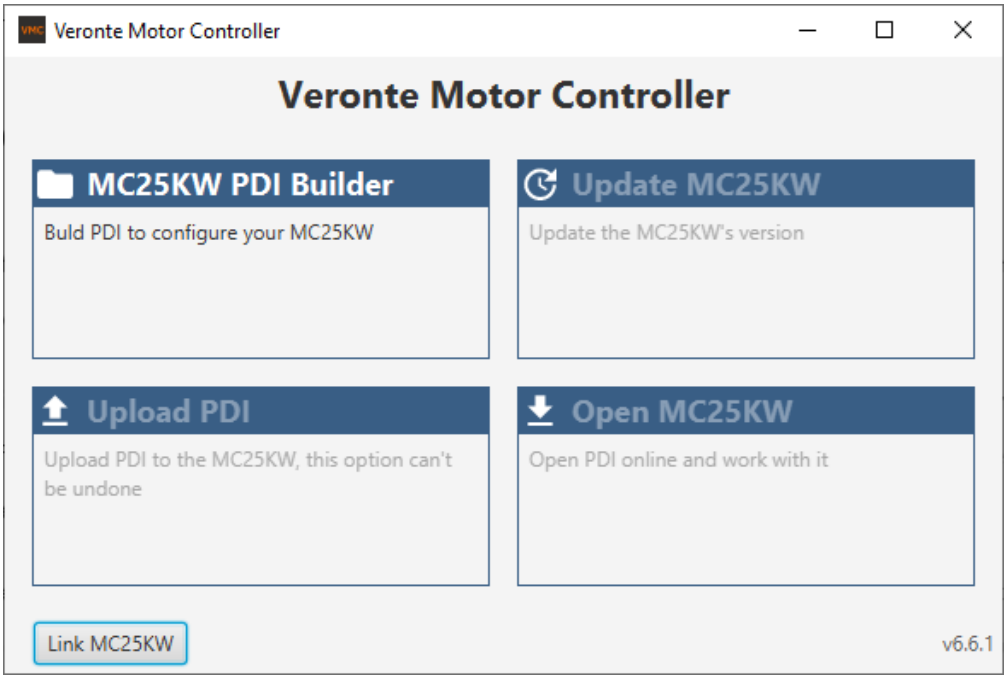
Then, type your MC280 address. Usually this is 35000 + your SN (which is written the box). For instance, if your SN is 0105, your ID will be **35105**.



If everything went well, a new MC280 will be displayed in the devices list. More MC280 units could be added following this instructions but the user is ready now to start configuring the motor controller using **MC280 PDI Builder**.

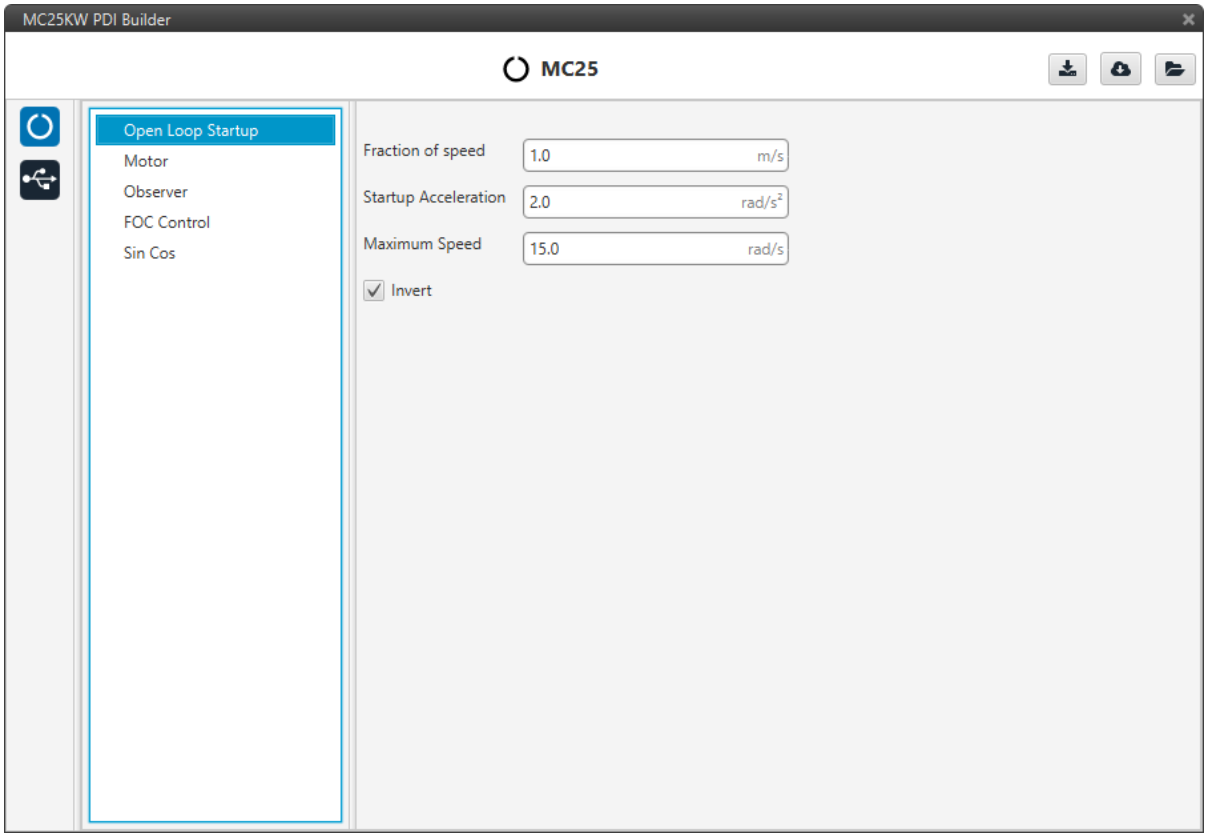
2.2 MC280 PDI Builder

After installing, MC280 PDI Builder the main menu will show as follows:



1. **MC280 PDI Builder**: Create a new PDI set of files to be saved and exported.
2. **Update MC280**: Update motor controller firmware version to a later one. Ask Embention support (support@embention.com) for firmware updates, future features or suggestions.
3. **Upload PDI**: Upload a previously created set of files to the current motor controller flash memory.
4. **Open MC280**: Edit the current MC280 settings.

At this point, the user can either create its own configuration offline by clicking on “MC280 PDI Builder” or start editing the one that is already loaded to the controller by clicking on “Open MC280”. The same menu will pop up:



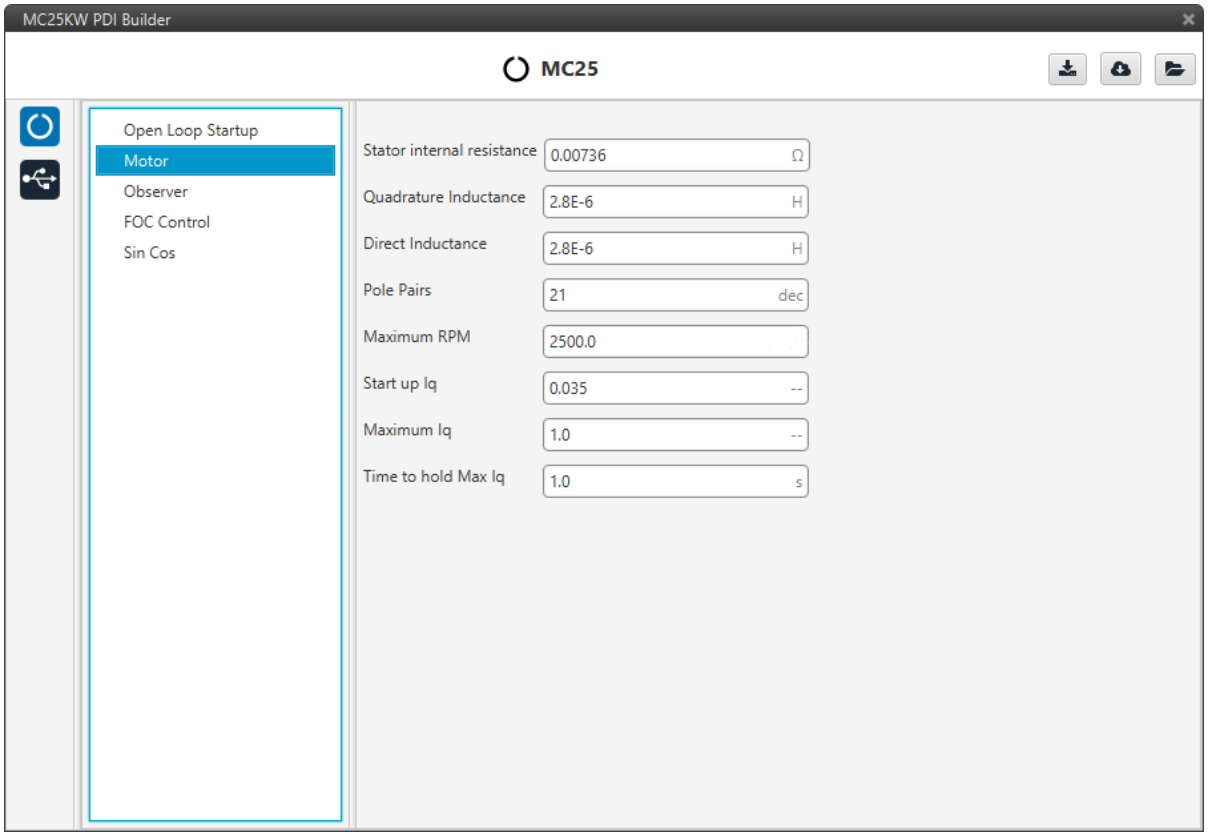
2.3 Open Loop Ramp

This menu sets the start ramp before engaging the closed loop control. There basically three parameters:

1. **Speed fraction**: Fraction of maximum speed (third field) to be reached before changing to closed loop.
2. **Startup Accelaration**: angular ancceleration to reach maximum speed. In rad/s^2.
3. **Maximum Speed**: Final speed to be reached before closing the control loop. In rad/s.
4. **Invert**: Change spining direction of the rotor. **This is the main way to invert the motor direction and will also affect the control and closed loop functionalities.**

2.4 Motor

This tab sets all the physical parameters of the motor that will be used with Veronte MC280. These are:



1. **Stator internal resistance:** This is the resistance that is usually specified in datasheets. Expressed in Ohms.
2. **Quadrature Inductance:** Usually called "Lq". Expressed in Henries (H).
3. **Direct Inductance:** Usually called "Ld". Expressed in Henries (H).
4. **Pole pairs:** Number of poles divided by two. For instance, if your motor has 42 poles, you must input 21 here.
5. **Maximum RPMs:** The maximmum RPMs the motor can reach in combination with your propeller. This parameter is only used to determine the equivalence between the maximum command (PWM or CAN) and the commanded RPM in speed closed loop mode.
6. **Startup Iq:** The current used to start the motor.
7. **Maximum Iq:** Maximum current the speed control loop can command to the quadrature current control loop. Please refer to FOC background section for further reference.
8. **Time to hold Max Iq:** Maximum time to hold the previous value before going to off.

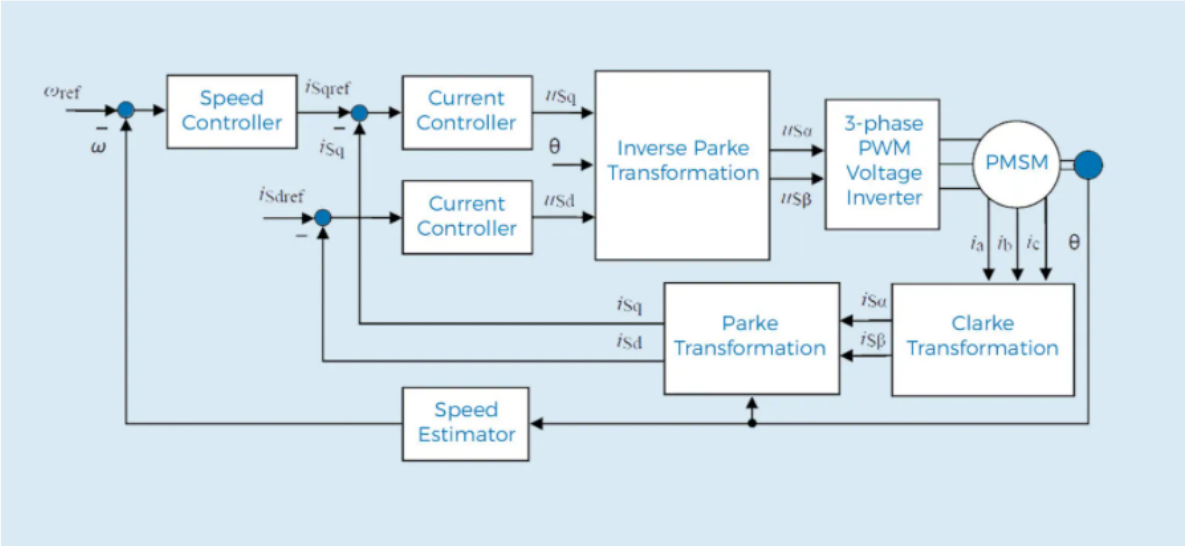
Warning: All the current are expressed in amperes/800. So if in order to input 35 Apk, **0.043** must be typed.

2.5 FOC (Field Oriented Control) Background

In order to achieve better dynamic performance, a more complex control scheme needs to be applied, to control the PM motor. With the mathematical processing power offered by the microcontrollers, we can implement advanced control strategies, which use mathematical transformations in order to decouple the torque generation and the magnetization functions in PM motors. Such de-coupled torque and magnetization control is commonly called rotor flux oriented control, or simply Field Oriented Control (FOC).

The Field Orientated Control consists of controlling the stator currents represented by a vector. This control is based on projections which transform a three phase time and speed dependent system into a two co-ordinate (d and q co-ordinates) time invariant system. These projections lead to a structure similar to that of a DC machine control. Field orientated controlled machines need two constants as input references: the torque component (aligned with the q co-ordinate) and the flux component (aligned with d co-ordinate). As Field Orientated Control is simply based on projections the control structure handles instantaneous electrical quantities. This makes the control accurate in every working operation (steady state and transient) and independent of the limited bandwidth mathematical model.

A basic scheme for the FOC is represented as follows:



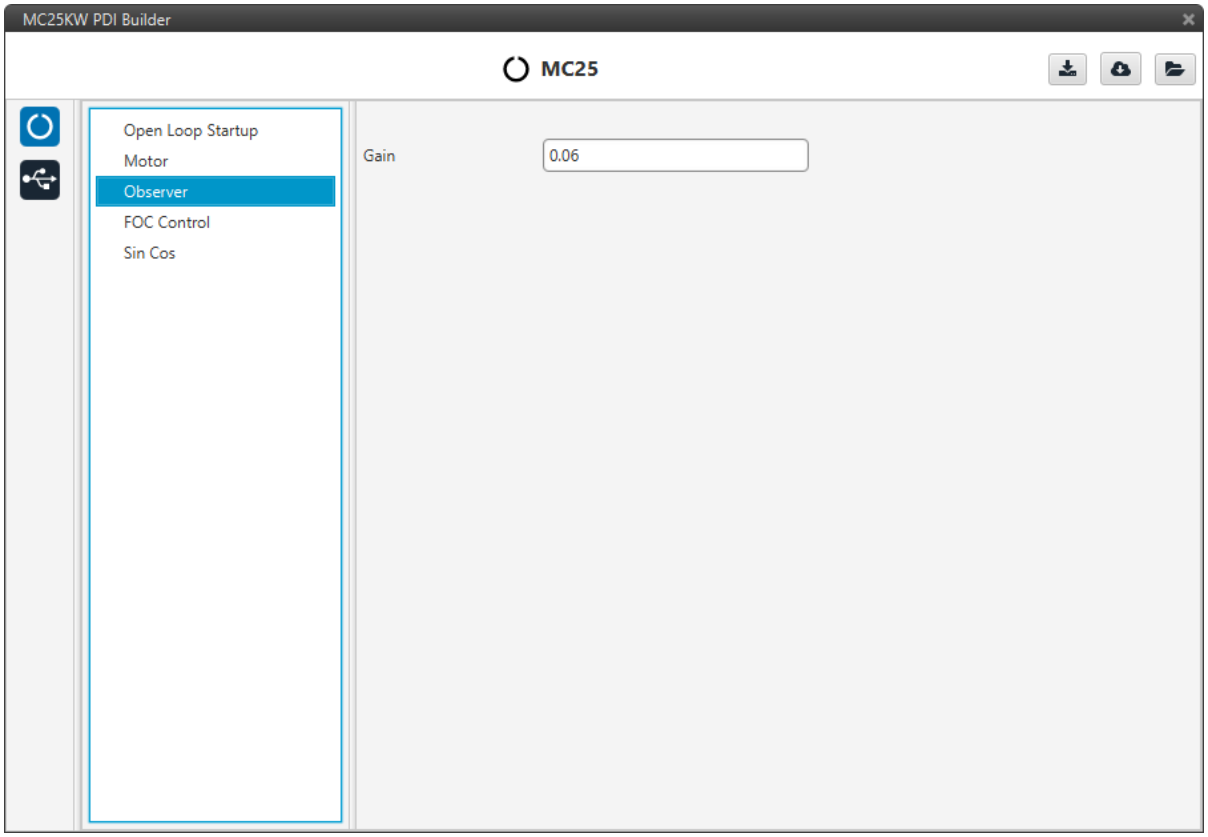
As it can seen, there some key pieces in this algorithm:

1. **Park/Clark transform**: These output a two co-ordinate time invariant system and a outputs a two co-ordinate time variant system respectively. As mentioned before, this is part of the process of getting two scalar values from a three phase time dependent system.
2. **PI Controllers**: There are three of them: two to control quadrature current and direct curent (torque and flux) and one to control speed. This last one is placed as the one that controls Iq PI (cascade control) which means that in order to get more speed the system will command more torque (or Iq).
3. **Speed estimator**: This block is able to estimate mechanical speed from current and voltage using the so-called "Sliding Mode Observer" algorithm.

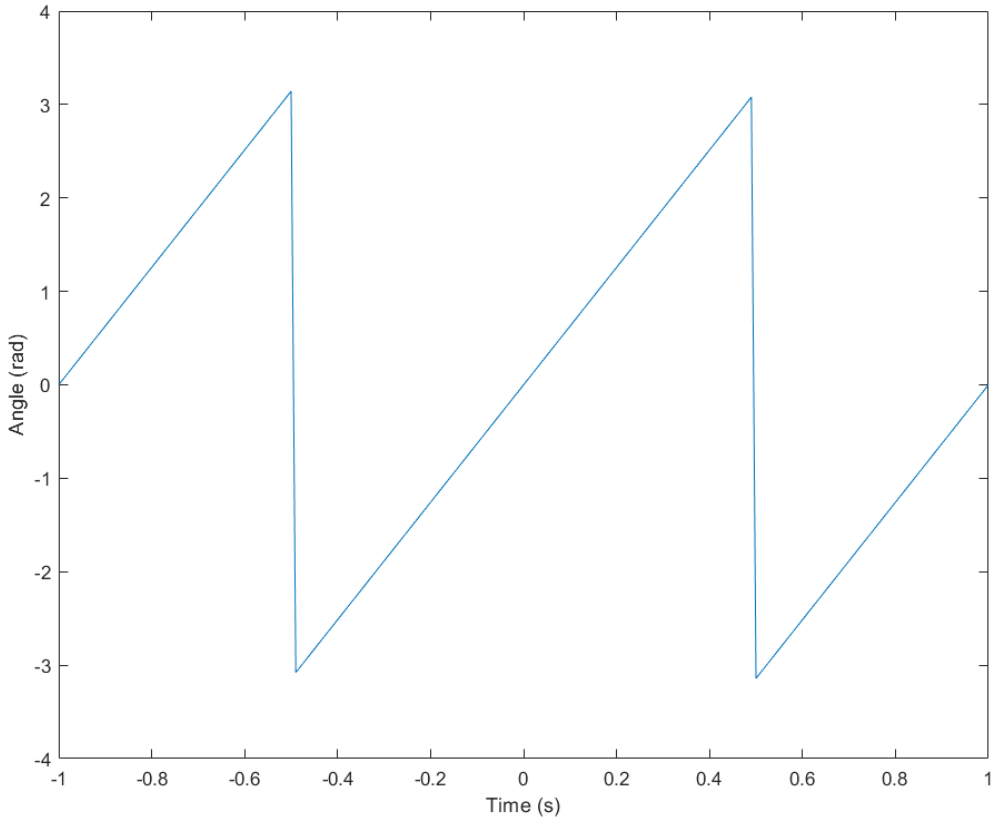
The main dificulty of this control proposal is to tune these three PI controller although in most cases both current PI are exactly the same due to PMSM properties. In addition, there is an extra gain that needs to be tuned as part of the angular speed estimator algorithm.

2.6 Sensorless RPM Observer/ Estimator

Once the basis of the FOC are covered, the rest of the features of this MC280 can be explained. This tab covers the observer parameters, which in this case is only one. The observer that is implemented has an ON/OFF controller that gets the estimated electrical angle. This electrical angle is later derived to find the angular speed. As it can be seen, here only this gain must be selected:



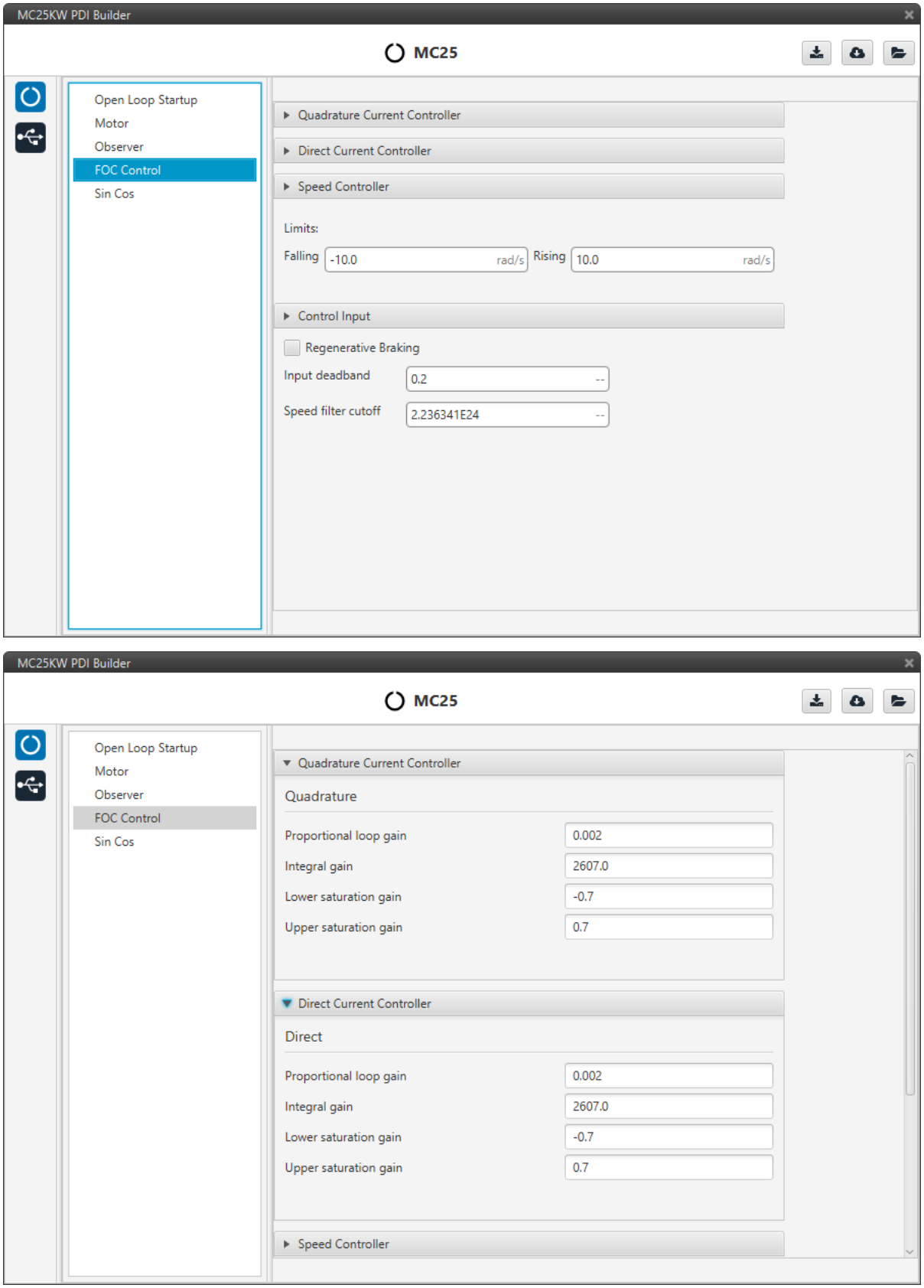
After this gain is properly tuned, the electrical angle should look like this:



Note: The algorithm includes an adaptive filter that automatically moves its cutoff frequency with the current mechanical angular speed, so the signal the user is monitoring is already filtered.

2.7 FOC Control

This is the main control menu, here all parameters regarding closed loop control are set. As mentioned before, the basic blocks that define the FOC control are three PI (Proportional, Integral controllers). First, both current PI are defined.



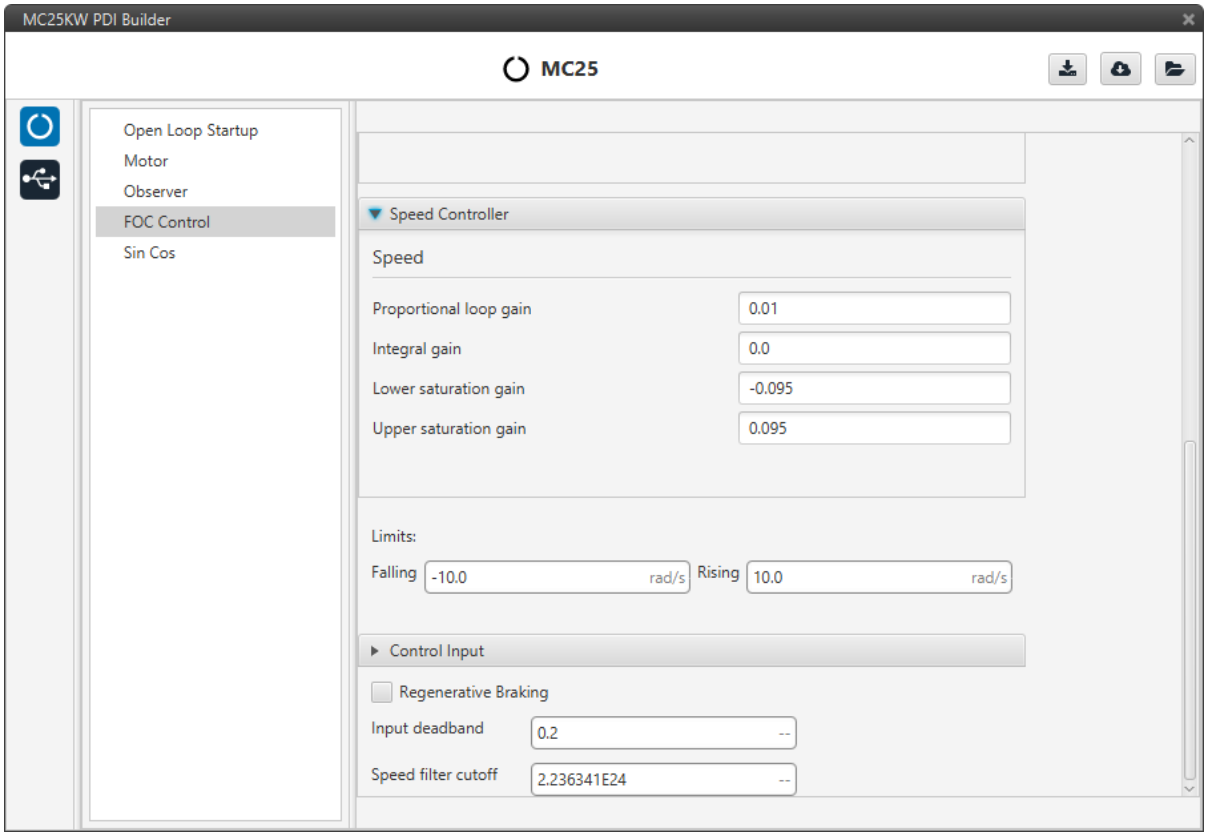
The form of the PI is the classical parallel form:

$$u = K_p \left(1 + \frac{1}{T_i} s\right)$$

Where integral gain refers to the quotient **1/Ti**. Lower and upper saturation gain are the limits at which the PI limit its output. In this case, as it is illustrated in FOC background, these outputs are **Vq** and **Vd**.

Note: All limits quantities are normalized, which means that are values between **0** and **1**.

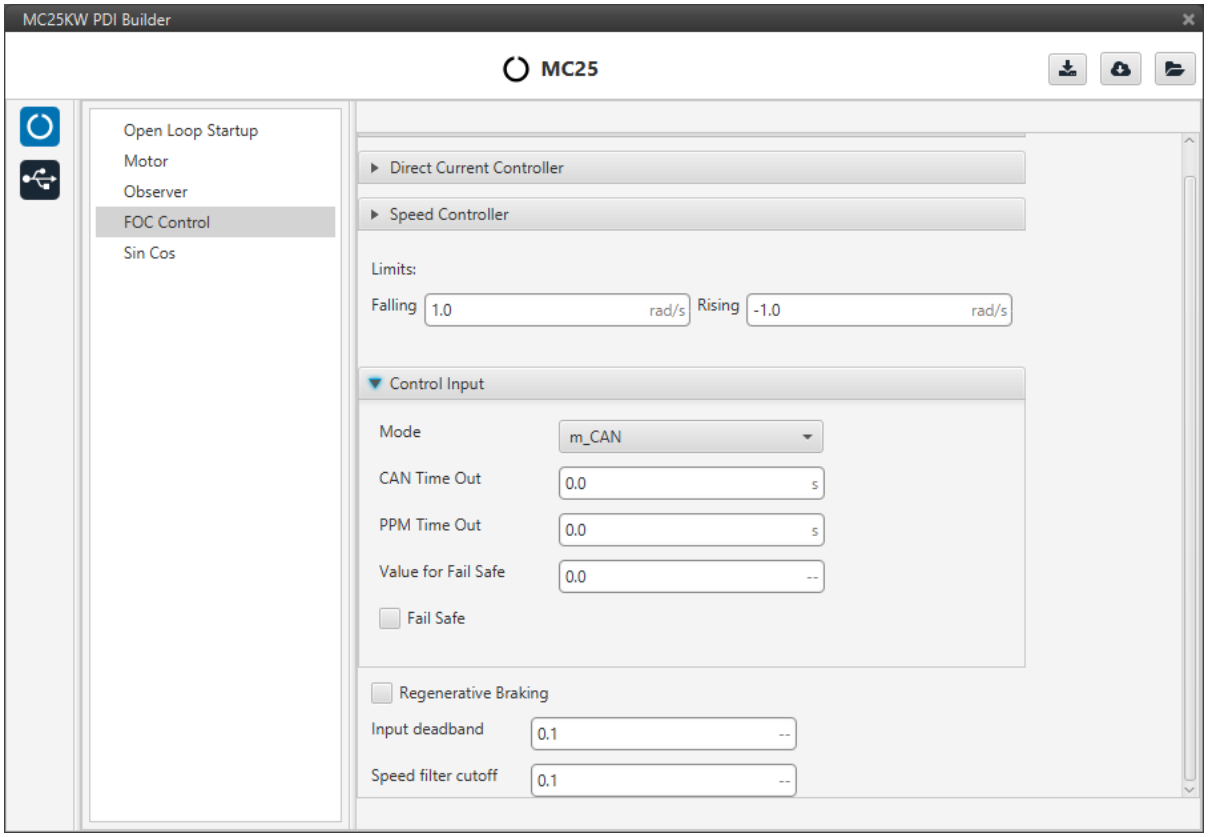
Equally, the speed controller can be set here as shown below. In this example this PI controller is limited to be output **76 A**.



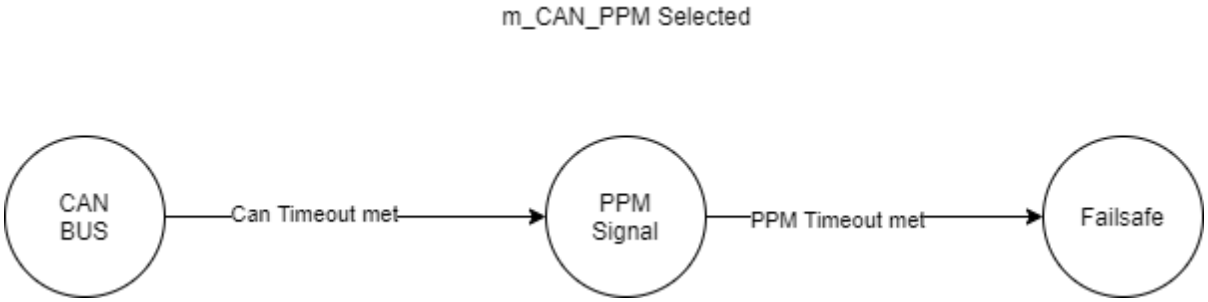
In addition, there is a rate limiter that could be used in case the slope needs to be limited with the input comand. Please note that is expressed in rad/s.

Control input allows the user to select the input source and several extra parameters such as:

1. **CAN Timeout:** Whenever this timeout is met, the control input will be changed to the next option if available. If none of them is available, it will end up in failsafe mode.
2. **PPM Timeout:** Same as before, but in this case the only option after this one is failsafe.
3. **Value for Fail Safe:** Value between 0 and 1. It will be written in case none of the previous options is available.



The input flow is the following in case the mode **m_CAN_PPM** is selected:



There some extra options than can be set under FOC control menu:

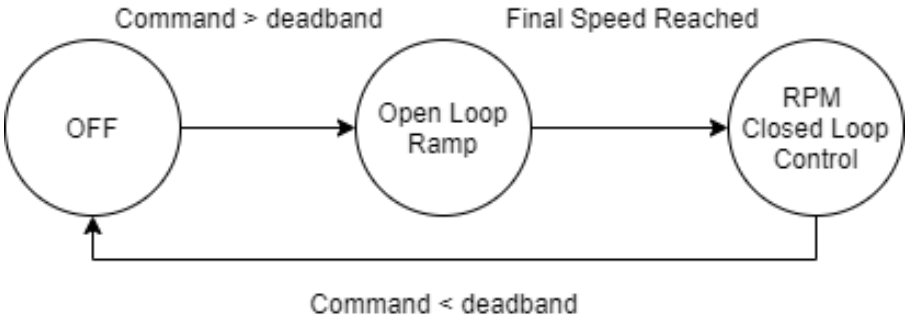
1. **Regenerative Braking:** Activates regnerative braking feature which is basically consisting of thoughing some current back to battery whenever the motor is loosing speed (it uses the kinetic energy to charge de battery). This not recommended in case the power side is not connected to a battery.
2. **Input deadband:** This is the value that defines when to start moving the motor. For example, it might be wanted to be different from zero in case an RC stick outputs around 0.1 of duty cycle by default.

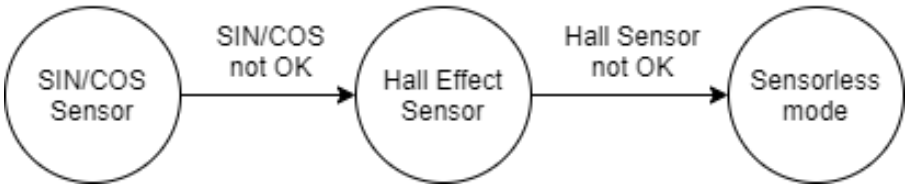
Note: In case CAN Bus is used to command (see CAN I/O section), this deadband can be calculated as **(desired deadband RPM)/(maximum RPM)**.

---

3. **Speed filter cutoff:** Cutoff frequency (in Hz) that will be applied to filter **Hall effect sensors**.

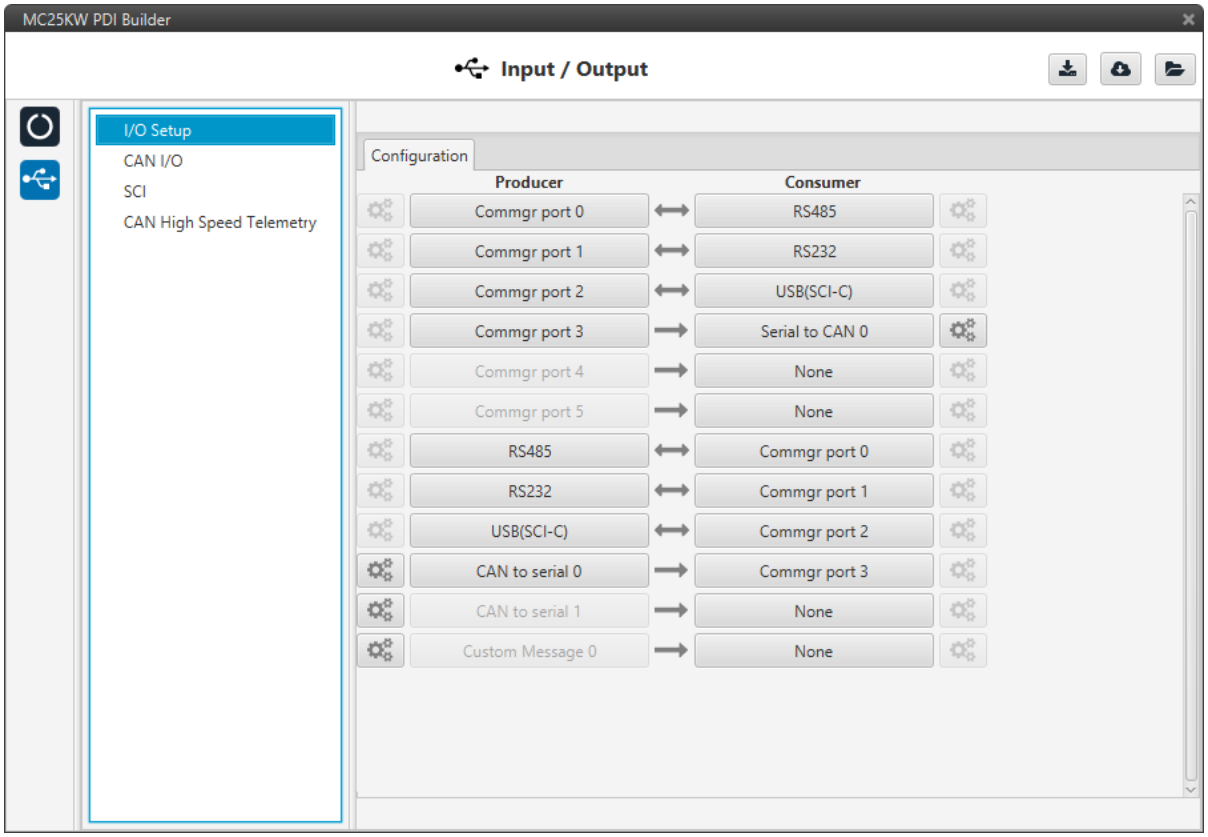
Finally, a state machine diagram is presented to clarify how the control and feeback sources work:





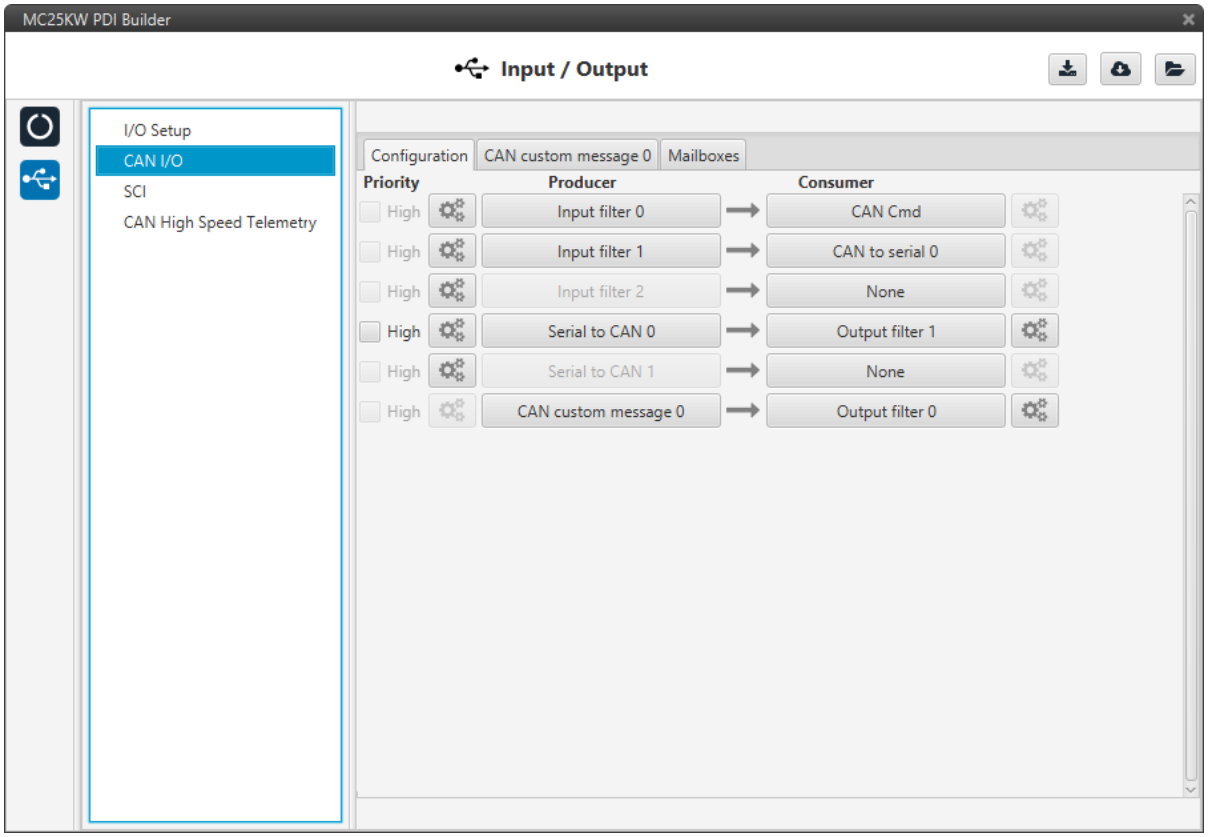
2.8 I/O Setup

This panel is used to stablish the mapping of ports and the duties they are performing. The layout and principles of function of this feature are common in all Veronte products, a complete explanantion can be found here .

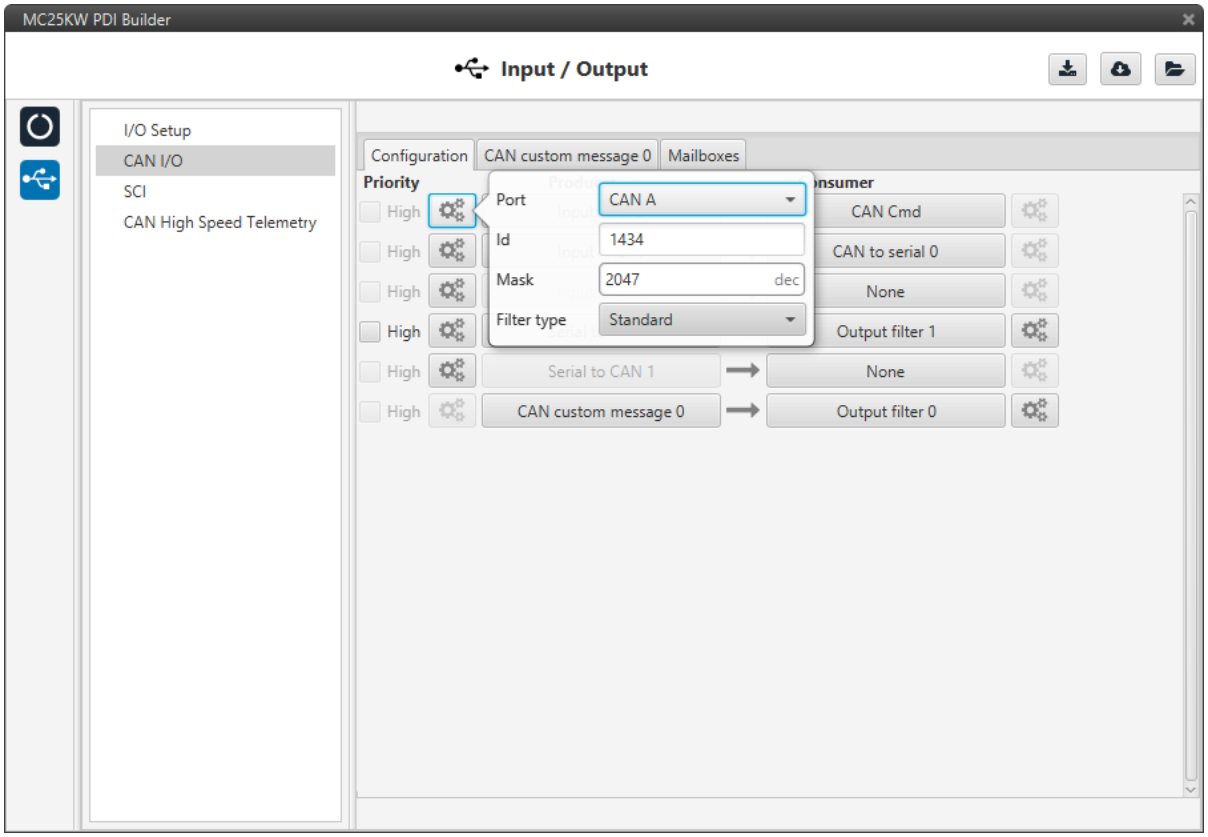


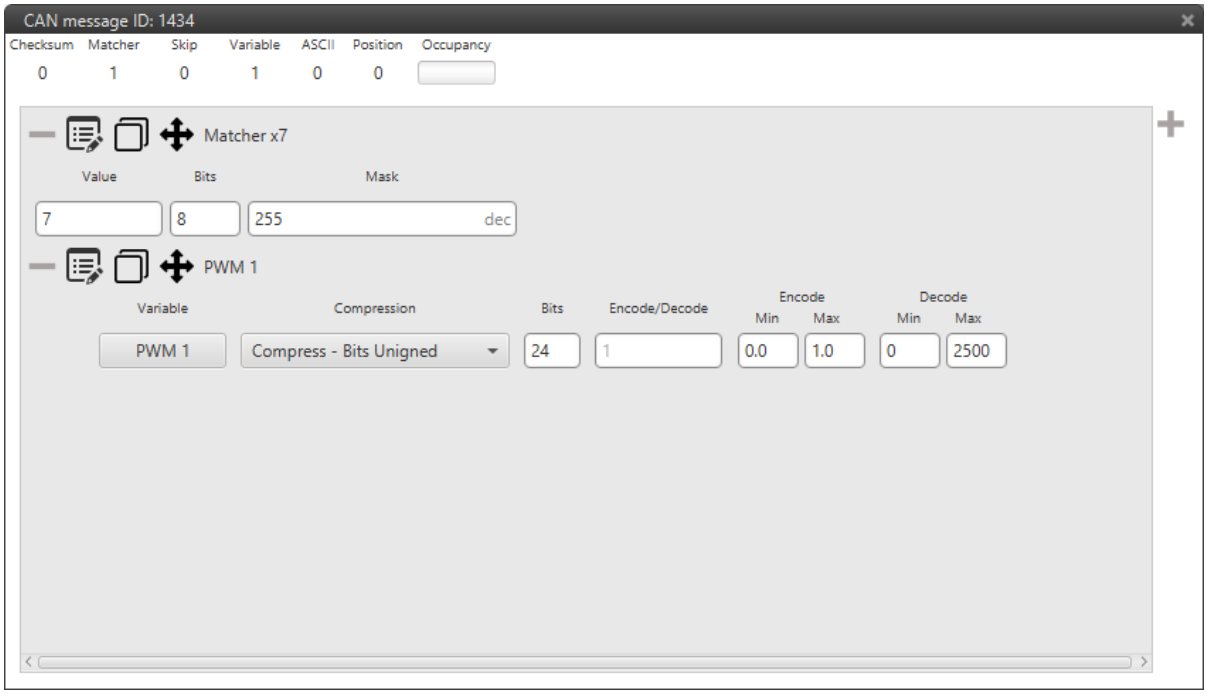
2.9 CAN I/O

This a common part that is shared with Veronte Autopilot products, please refer to this page for further information.



Altough, most of this menu could be familiar to veronte users, there is one new feature : **CAN CMD**. This is the input ESC expects to command the motor. By default, it uses the ID **1434 (standard)** and the message structure is the following:



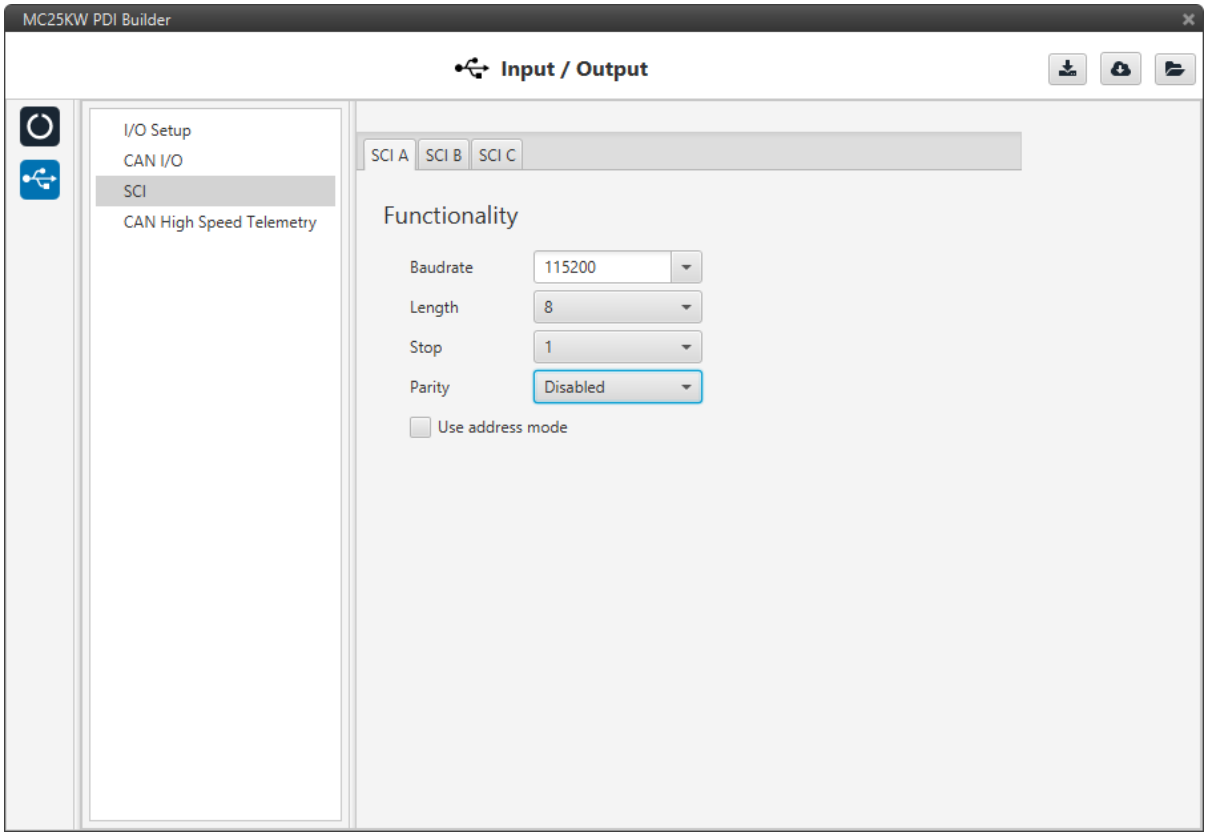


Warning: Note that what is represented as PWM1 is in fact sent as a value between 0 and 2500 as a result of the decoding factor. This last 24 bit value is in fact an RPM value.

2.10 SCI

The following fields can be configured:

1. **Baud rate:** This field specifies how fast data is sent over a serial line.
2. **Length:** This field defines the number of data bits in each character.
3. **Stop:** Stop bits sent at the end of every character.
4. **Parity:** is a method of detecting errors in transmission. When parity is used with a serial port, an extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit, is always odd or always even.



Note: MC280 is running at 10kHz.